Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in this Application:

Listing of Claims:

1. (Currently amended) A method for permuting two dimensional (2-D) data in a programmable processor comprising the steps of:

decomposing said two dimensional data into at least one atomic element said two dimensional data being located in at least one source register said at least one atomic element of said two dimensional data is a 2x2 matrix and said two dimensional data is decomposed into data elements in said matrix; ~~and~~

determining at least one permutation instruction for rearrangement of said data in said atomic element;

said data elements being rearranged by said at least one permutation instruction, each of said data elements representing a subword having one or more bits; and

applying said permutation instructions to said subwords and placing said permutated subwords into a designation register.

2. (Canceled).

3. (Currently amended) The method of claim ~~2~~ 1 further comprising a triangle in said matrix, said data elements in said triangle being rearranged by said at least one permutation instruction.

4. (Currently amended) The method of claim ~~2~~ 1 wherein said permutation instruction swaps a first one of said data elements and a second one of said data elements, said first one of said data elements and said second one of said data elements being in the same column of said matrix.

5. (Currently amended) The method of claim ~~2~~ 1 wherein said permutation instruction swaps a first one of said data elements and a second one of said data elements, said first one of said data elements and said second one of said data elements being in the same row of said matrix.

6. (Currently amended) The method of claim ~~2~~ 1 wherein said permutation instruction swaps a first one of said data elements and a second one of said data elements, said first one of said data elements and said second one of said data elements being diagonal to one another in said matrix.

7. (Currently amended) The method of claim ~~2~~ 1 wherein said permutation instruction rotates a first one of said data elements by one or more positions in said matrix.

8. (Original) The method of claim 3 wherein said permutation instruction rotates a first one of said data elements by one or more positions in said triangle.

9. (Currently amended) The method of claim 1 wherein said programmable processor is a microprocessor, digital signal processor, media processor, multimedia processor, cryptographic processor or programmable_System-On-Chip (SOC).

10. (Currently amended) The method of claim ~~2~~ 1 wherein said permutation instruction alternately selects a first subword from a first column of said matrix and a second subword from said first column of said matrix and swaps the selected said first subword and the selected said second subword.

11. (Currently amended) The method of claim ~~2~~ 1 wherein said permutation instruction swaps a first subword in a first row of said matrix with a second subword in said first row of said matrix.

12. (Currently amended) The method of claim ~~2~~ 1 wherein said permutation instruction alternately selects a first subword from a first column of said matrix and a second subword from said first column of said matrix, swaps the selected said first subword and the selected said second subword and swaps the swapped first subword in a first row of said matrix with a third subword in said first row of said matrix or the swapped second subword in a second row of said matrix with a fourth subword in said second row of said matrix.

13. (Currently amended) The method of claim ~~2~~ 1 wherein said permutation instruction conditionally selects a first subword from a first column of said matrix and a second subword from said first column of said matrix dependant on a permutation control bit and swaps the selected said first subword and the selected said second subword.

14. (Currently amended) The method of claim ~~2~~ 1 wherein said permutation instruction conditionally swaps a first subword in a first row of said matrix with a second subword in said first row of said matrix dependant on a permutation control bit.

15. (Currently amended) The method of claim ~~2~~ 1 wherein said permutation instruction conditionally selects a first subword from a first column of said matrix and a second subword from said first column of said matrix dependant on a permutation control bit, swaps the selected said first subword and the selected said second subword and conditionally swaps the swapped first subword in a first row of said matrix with a third subword in said first row of said matrix or the swapped second subword in a second row of said matrix with a fourth subword in said second row of said matrix dependant on a permutation control bit.

16. (Currently amended) The method of claim ~~2~~ 1 wherein said permutation instruction defines a size of said subword, defines a subset of subwords in said sequence of subwords,

swaps a first subword in said subset with a second subword in said subset and concatenates the swapped first subword and second subword.

17. (Currently amended) The method of claim ~~2~~ 1 wherein said permutation instruction conditionally concatenates one or more odd elements of a first said subword sequentially with one or more second odd elements of a second said subword.

18. (Original) The method of claim 17 wherein said odd elements of a first said subword and odd elements of a second said subword are 32-bit subwords, 16-bit subwords or 8-bit subwords and said first subword and said second subword are 64-bit subwords.

19. (Currently amended) The method of claim ~~2~~ 1 wherein said permutation instruction conditionally concatenates one or more first even elements of a first said subword sequentially with one or more second even elements of a second said subword.

20. (Original) The method of claim 19 wherein said even elements of said first said subword and said even elements of said second said subword are 32-bit subwords, 16-bit subwords or 8-bit subwords and said first subword and said second subword are 64-bit subwords.

21. (Original) The method of claim 1 wherein said permutation instructions for said atomic unit is defined for larger subword sizes at successively higher hierarchical levels.

22. (Currently amended) A system for permuting two-dimensional (2-D) data in a programmable processor comprising:

at least one source register containing said two dimensional data;

means for decomposing said two dimensional data into at least one atomic element said at least one atomic element of said two dimensional data is a 2x2 matrix; ~~and~~

a destination register;

means for determining at least one permutation instruction for rearrangement of said data in said atomic element said two dimensional data is decomposed into data elements in said matrix, said data elements being rearranged by said at least one permutation instruction, each of said data elements representing a subword having one or more bits; and

means for placing said permutated subwords into said destination register.

23. (Canceled).

24. (Currently amended) The system of claim ~~23~~ 22 further comprising a triangle in said matrix, said data elements in said triangle being rearranged by said at least one permutation instruction.

25. (Currently amended) The system of claim of claim ~~23~~ 22 wherein said permutation instruction swaps a first one of said data elements and a second one of said data elements, said first one of said data elements and said second one of said data elements being in the same column of said matrix.

26. (Currently amended) The system of claim ~~23~~ 22 wherein said permutation instruction swaps a first one of said data elements and a second one of said data elements, said first one of said data elements and said second one of said data elements being in the same row of said matrix.

27. (Currently amended) The system of claim ~~23~~ 22 wherein said permutation instruction swaps a first one of said data elements and a second one of said data elements, said first one of said data elements and said second one of said data elements being diagonal to one another in said matrix.

28. (Currently amended) The system of claim ~~23~~ 22 wherein said permutation instruction rotates a first one of said data elements by one or more positions is said matrix.

29. (Original) The system of claim 24 wherein said permutation instruction rotates a first one of said data elements by one or more positions in said triangle.

30. (Currently amended) The system of claim ~~23~~ 22 wherein said permutation instruction conditionally selects a first subword from a first column of said matrix and a second subword from said first column of said matrix dependant on a permutation control bit and swaps the selected said first subword and the selected said second subword.

31. (Currently amended) The system of claim ~~23~~ 22 wherein said permutation instruction conditionally swaps a first subword in a first row of said matrix with a second subword in said first row of said matrix dependant on a permutation control bit.

32. (Currently amended) The system of claim ~~23~~ 22 wherein said permutation instruction conditionally selects a first subword from a first column of said matrix and a second subword from said first column of said matrix dependant on a permutation control bit, swaps the selected said first subword and the selected said second subword and conditionally swaps the swapped first subword in a first row of said matrix with a third subword in said first row of said matrix or the swapped second subword in a second row of said matrix with a fourth subword in said second row of said matrix dependant on a permutation control bit.

33. (Currently amended) The system of claim ~~23~~ 22 wherein said permutation instruction defines a size of said subword, defines a subset of subwords in said sequence of subwords,

swaps a first subword in said subset with a second subword in said subset and concatenates the swapped first subword and second subword.

34. (Currently amended) The system of claim ~~23~~ 22 wherein said permutation instruction conditionally concatenates one or more odd elements of a first said subword sequentially with one or more second odd elements of a second said subword.

35. (Currently amended) The system of claim ~~23~~ 22 wherein said odd elements of said first said subword and said odd elements of said second said subword are 32-bit subwords, 16-bit subwords or 8-bit subwords and said first subword and said second subword are 64-bit subwords.

36. (Currently amended) The system of claim ~~23~~ 22 wherein said permutation instruction conditionally concatenates one or more first even elements of a first said subword sequentially with one or more second even elements of a second said subword.

37. (Currently amended) The system of claim ~~23~~ 22 wherein said even elements of said first said subword and said ~~evene~~ even elements of said second said subword are 32-bit subwords, 16-bit subwords or 8-bit subwords and said first subword and said second subword are 64-bit subwords.

38. (Currently amended) The system of claim ~~23~~ 22 wherein said programmable processor is a microprocessor, digital signal processor, media processor, multimedia processor, cryptographic processor or programmable System-On-Chip (SOC).

39. (Currently amended) The system of claim ~~23~~ 22 wherein said permutation instructions for said atomic unit is defined for larger subword sizes at successively higher hierarchical levels.

40. (Currently amended) A method for performing subword permutations in a programmable processor comprising the steps of:

in response to a permutation instruction alternately selecting a first subword from a first sequence of subwords in a first register and a second subword from a second sequence of subwords in a second register; and

concatenating the selected said first subword and the selected said second subword into a third sequence of subwords in a third register.

41. (Original) The method of claim 40 further comprising the step of repeating said alternately selecting step for each of said subwords in said first sequence of subwords and each of said subwords in said second sequence of subwords.

42. (Original) The method of claim 40 wherein said permutation instruction comprises a parameter for determining the number of bits in said first subword and said second subword to be selected, a reference to a first source register which contains said first sequence of subwords, a reference to a second source register which contains said second sequence of subwords and optionally a reference to a destination register which contains said third sequence of subwords.

43. (Currently amended) The method of claim 40 wherein each subword comprises one or more bits.

44. (Currently amended) A method for performing subword permutation in a programmable processor comprising the steps of:

swapping a first subword in a first register with a second subword in a sequence of subwords in a second register and concatenating the swapped said first subword and said second subword into a second sequence of subwords in a third register.

45. (Original) The method of claim 44 further comprising the step of repeating said swapping step for each of said subwords in said sequence of subwords.

46. (Original) The method of claim 44 wherein said permutation instruction comprises a parameter for determining the number of bits in said first subword and said second subword to be swapped, a reference to a source register which contains said sequence of subwords and optionally a reference to a destination register which contains said second sequence of subwords.

47. (Original) The method of claim 44 wherein each subword comprises one or more bits.

48. (Currently amended) A method for performing subword permutation in a programmable processor comprising the steps of:

in response to a permutation instruction alternately selecting a first subword from a first sequence of subwords in a first register and a second subword from a second sequence of subwords in a second register;

concatenating the selected said first subword and the selected said second subword into a third sequence of subwords in a third register;

swapping a third subword in said third sequence of subwords with a fourth subword in said second sequence or said third sequence of subwords; and

concatenating the swapped said third subword with the swapped said fourth subword into a fourth sequence of subwords.

49. (Original) The method of claim 48 further comprising the step of repeating said alternately selecting step for each of said subwords in said first sequence of subwords and repeating said swapping step for each of said subwords in said third sequence of subwords.

50. (Original) The method of claim 48 wherein said permutation instruction comprises a parameter for determining the number of bits to be selected and to be swapped, a reference to a first source register which contains said first sequence of subwords, a reference to a second source register which contains said second sequence of subwords and optionally a reference to a destination register which contains said third sequence of subwords or said fourth sequence of subwords.

51. (Original) The method of claim 48 wherein each subword comprises one or more bits.

52. (Original) A method for performing subword permutations in a programmable processor comprising the steps of:

in response to a permutation instruction conditionally alternately selecting a first subword from a first sequence of subwords and a second subword from a second sequence of subwords dependant on permutation control bits; and

concatenating the selected said first subword and the selected said second subword into a third sequence of subwords.

53. (Original) The method of claim 52 further comprising the step of repeating said conditionally selecting step for each of said subwords in said first sequence of subwords and each of said subwords in said second sequence of subwords.

54. (Original) The method of claim 52 wherein said permutation instruction comprises a control bit configuration for determining said permutation control bits, a first source register which contains said first sequence of subwords, a reference to a second source register which contains said second sequence of subwords and optionally a reference to a destination register which contains said third sequence of subwords.

55. (Currently amended) The method of claim 52 wherein each subword comprises one or more bits.

56. (Currently amended) A method for performing subword permutation in a programmable processor comprising the steps of:

conditionally swapping a first subword with a second subword in a sequence of subwords dependant on permutation control bits in a first register and concatenating the swapped said first subword and said second subword into a second sequence of subwords in a second register.

57. (Original) The method of claim 56 further comprising the step of repeating said conditionally swapping step for each of said subwords in said sequence of subwords.

58. (Original) The method of claim 56 wherein said permutation instruction comprises a control bit configuration for determining said permutation control bits, a reference to a source register which contains said sequence of subwords and optionally a reference to a destination register which contains said second sequence of subwords.

59. (Original) The method of claim 56 wherein each subword comprises one or more bits.

60. (Currently amended) A method for performing subword permutation in a programmable processor comprising the steps of:

in response to a permutation instruction conditionally electing a first subword from a first sequence of subwords in a first register and a second subword from a second sequence of subwords in a second register dependant on permutation control bits;

concatenating the selected said first subword and the selected said second subword into a third sequence of subwords in a third register;

conditionally swapping a third subword in said third sequence of subwords with a fourth subword in said second sequence or said third sequence of subwords dependant on said permutation control bits; and

concatenating the swapped said third subword with the swapped said fourth subword into a fourth sequence of subwords.

61. (Original) The method of claim 60 further comprising the step of repeating said conditionally selecting step for each of said subwords in said first sequence of subwords and repeating said conditionally swapping step for each of said subwords in said third sequence of subwords.

62. (Original) The method of claim 60 wherein said permutation instruction comprises a control bit configuration for determining said permutation control bits, a reference to a first source register which contains said first sequence of subwords, a reference to a second source register which contains said second sequence of subwords and optionally a reference to a destination register which contains said third sequence of subwords or said fourth sequence of subwords.

63. (Currently amended) The method of claim 60 wherein each subword comprises one or more bits.

64. (Currently amended) A method for performing subword permutation of a sequence of subwords in a programmable processor comprising the steps of:

defining a size of said subword;

defining a subset of subwords in said sequence of subwords;

swapping a first subword in said subset in a first register with a second subword in a sequence of subwords in a second register and concatenating the swapped first subword and second subword into a second sequence of subwords in a third register; and

repeating said swapping step for consecutive subsets of subwords.

65. (Original) The method of claim 64 wherein said permutation instruction comprises a parameter for indicating said size of said subword, a parameter for indicating a number of elements in each said subset; a parameter for indicating permutation configuration bits, a source register which contains said first sequence of subwords and optionally a reference to a destination register which contains said second sequence of subwords.

66. (Original) The method of claim 64 wherein each subword comprises one or more bits.

67. (Currently amended) A method for performing subword permutation in a programmable processor comprising the steps of:

in response to a permutation instruction conditionally concatenating one or more odd elements of a first said subword in a first register sequentially with one or more second odd elements of a second said subword in a second register.

68. (Original) The method of claim 67 wherein said odd elements of said first said subword and said odd elements of said second said subword are 32-bit subwords, 16-bit subwords or 8-bit subwords and said first subword and said second subword are 64-bit subwords.

69. (Currently amended) A method for performing subword permutation in a programmable processor comprising the steps of:

in response to a permutation instruction conditionally concatenating one or more first even elements of a first said subword in a first register sequentially with one or more second even elements of a second said subword in a second register.

70. (Original) The method of claim 69 wherein said even elements of said first said subword and said even elements of said second said subword are 32-bit subwords, 16-bit subwords or 8-bit subwords and said first subword and said second subword are 64-bit subwords.

71. (Currently amended) A system for performing subword permutations in a programmable processor comprising:

in response to a permutation instruction, means for alternately selecting a first subword from a first sequence of subwords in a first register and a second subword from a second sequence of subwords in a second register; and

means for concatenating the selected said first subword and the selected said second subword into a third sequence of subwords in a third register.

72. (Original) The system of claim 71 further comprising means for repeating said means for alternately selecting a first subword for each of said subwords in said first sequence of subwords and each of said subwords in said second sequence of subwords.

73. (Original) The system of claim 71 wherein said permutation instruction comprises a parameter for determining the number of bits in said first subword and said second subword to be selected, a reference to a first source register which contains said first sequence of subwords, a reference to a second source register which contains said second sequence of subwords and optionally a reference to a destination register which contains said third sequence of subwords.

74. (Currently amended) A system for performing subword permutation in a programmable processor comprising:

means for swapping a first subword in a first register with a second subword in a sequence of subwords in a second register and concatenating the swapped said first subword and said second subword into a second sequence of subwords in a third register.

75. (Original) The system of claim 74 further comprising means for repeating said means for swapping for each of said subwords in said sequence of subwords.

76. (Original) The system of claim 74 wherein said permutation instruction comprises a parameter for determining the number of bits in said first subword and said second subword to be swapped, a reference to a source register which contains said sequence of subwords and optionally a reference to a destination register which contains said second sequence of subwords.

77. (Currently amended) A system for performing subword permutation in a programmable processor comprising:

in response to a permutation instruction, means for alternately selecting a first subword from a first sequence of subwords in a first register and a second subword from a second sequence of subwords in a second register;

means for concatenating the selected said first subword and the selected said second subword into a third sequence of subwords in a third register;

means for swapping a third subword in said third sequence of subwords with a fourth subword in said second sequence or said third sequence of subwords; and

means for combining the said third sequence of subwords with the swapped said fourth subword into a fourth sequence of subwords.

78. (Original) The system of claim 77 further comprising means for repeating said means for alternately selecting for each of said subwords in said first sequence of subwords and repeating said means for swapping for each of said subwords in said second or third sequence of subwords.

79. (Original) The system of claim 77 wherein said permutation instruction comprises a parameter for determining the number of bits to be selected and to be swapped, a reference to a first source register which contains said first sequence of subwords, a reference to a second source register which contains said second sequence of subwords and optionally a

reference to a destination register which contains said third sequence of subwords or said fourth sequence of subwords.

80. (Currently amended) A system for performing subword permutations in a programmable processor comprising the steps of:

in response to a permutation instruction means for conditionally selecting a first subword from a first sequence of subwords in a first register and a second subword from a second sequence of subwords in a second register dependant on permutation control bits; and

means for concatenating the selected said first subword and the selected said second subword into a third sequence of subwords in a third register.

81. (Original) The system of claim 80 further comprising means for repeating said means for conditionally selecting for each of said subwords in said first sequence of subwords and each of said subwords in said second sequence of subwords.

82. (Currently amended) The system of claim 80 wherein said permutation instruction comprises a control bit configuration for determining said permutation control bits, a first source register which contains said first sequence of subwords, a reference to a second source register which contains ~~siad~~ said second sequence of subwords and optionally a reference to a destination register which contains said third sequence of subwords.

83. (Currently amended) A system for performing subword permutation in a programmable processor comprising:

in response to a permutation instruction, means for conditionally swapping a first subword in a first register with a second subword in a sequence of subwords in a second register dependant on permutation control bits and concatenating the swapped said first subword and said second subword into a second sequence of subwords in a third register.

84. (Original) The system of claim 83 further comprising means for repeating said means for conditionally swapping for each of said subwords in said sequence of subwords.

85. (Original) The system of claim 84 wherein said permutation instruction comprises a control bit configuration for determining said permutation control bits, a reference to a source register which contains said sequence of subwords and optionally a reference to a destination register which contains said second sequence of subwords.

86. (Currently amended) A system for performing subword permutation in a programmable processor comprising:

in response to a permutation instruction, means for conditionally selecting a first subword from a first sequence of subwords in a first register and a second subword from a second sequence of subwords in a second register dependant on permutation control bits;

means for concatenating the selected said first subword and the selected said second subword into a third sequence of subwords;

means for conditionally swapping a third subword in said third sequence of subwords with a fourth subword in said second sequence or said third sequence of subwords dependant on said permutation control bits in a third register; and

means for combining the third sequence of subwords with the swapped said fourth subword into a fourth sequence of subwords.

87. (Currently amended) The system of claim 86 further comprising means for repeating said means for conditionally selecting each of said subwords in said first sequence of subwords and repeating said means for conditionally swapping for each of said subwords in said ~~seond~~ second or third sequence of subwords.

88. (Original) The system of claim 86 wherein said permutation instruction comprises a control bit configuration for determining said permutation control bits, a reference to a first source register which contains said first sequence of subwords, a reference to a second source register which contains said second sequence of subwords and optionally a reference to a destination register which contains said third sequence of subwords or said fourth sequence of subwords.

89. (Original) A system for performing subword permutation of a sequence of subwords in a programmable processor comprising:

means for defining a size of said subword;

means for defining a subset of subwords in said sequence of subwords;

means for swapping a first subword in said subset in a first register with a second subword in a sequence of subwords in a second register and concatenating the swapped first subword and second subword into a second sequence of subwords; and means for repeating said swapping step for consecutive subsets of subwords.

90. (Original) The system of claim 89 wherein said permutation instruction comprises a parameter for indicating said size of said subword, a parameter for indicating a number of elements in each said subset; a parameter for indicating permutation configuration bits, a source register which contains said first sequence of subwords and optionally a reference to a destination register which contains said second sequence of subwords.

91. (Currently amended) A system for performing subword permutation in a programmable processor comprising:

in response to a permutation instruction conditionally concatenating one or more odd elements of a first said subword in a first register sequentially with one or more second odd elements of a second said subword in a second register.

92. (Original) The system of claim 91 wherein said odd elements of said first said subword and said odd elements of said second said subword are 32-bit subwords, 16-bit subwords or 8-bit subwords and said first subword and said second subword are 64-bit subwords.

93. (Currently amended) A system for performing subword permutation in a programmable processor comprising:

in response a permutation instruction conditionally concatenating one or more first even elements of a first said subword in a first register sequentially with one or more second even elements of a second said subword in a second register.

94. (Original) The system of claim 93 wherein said odd elements of said first said subword and said odd elements of said second said subword are 32-bit subwords, 16-bit subwords or 8-bit subwords and said first subword and said second subword are 64-bit subwords.